Lineaarinen ohjelmointi ja sen opetus toisen asteen oppilaitoksissa

Leo Seppälä

20. tammikuuta 2020

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

Tiedekunta – Fakultet – Faculty Matemaattis-luonnontieteellinen tiedekunta	Koulutusohjelma – Utbildningsprogram – Degree programme Matematiikan opettajan maisteriohjelma	
Tekijä – Författare – Author Leo Valtteri Seppälä		
Työn nimi – Arbetets titel – Title Lineaarinen ohjelmointi ja sen opetus toisen asteen oppilaitoksissa		
Työn laji – Arbetets art – Level Pro gradu -tutkielma	Aika – Datum – Month and year Tammikuu 2020	Sivumäärä – Sidoantal – Number of pages 48

Tiivistelmä – Referat – Abstract

Lineaarinen ohjelmointi on matemaattista optimointia, jota voidaan hyödyntää monissa käytännön ongelmissa. Tässä tutkielmassa käydään läpi lineaarisen ohjelmoinnin ongelmia ja niiden ratkaisutapoja, sekä miten lineaarista ohjelmointia voisi esitellä lukio-opetuksessa.

Lineaarinen ohjelmoinnin ongelma muodostuu lineaarisista ehdoista ja objektifunktiosta. Lineaarisen ohjelmoinnin ongelmilla on olemassa yleinen muoto, joka auttaa ongelman käsittelyä eri ratkaisumenetelmillä.
Lineaarisessa ohjelmoinnissa pitää ongelmasta rakentaa matemaattinen malli. Tämä malli voidaan rakentaa sarakemenetelmällä. Mallin rakennusprosessi jatkuu järjestelmällisesti, kunnes kaikki ongelman tarpeelliset tiedot on muutettu sellaiseen muotoon, että ongelmaa voidaan käsitellä lineaarisen ohjelmoinnin työkaluilla

Tutkielmassa on todistus sille, että lineaarisessa ohjelmoinnissa ongelman suurin ja pienin arvo löytyvät ehtolauseiden rajaamaan alueiden ääripisteistä. Tätä tietoa hyödyntämällä yksinkertaisia lineaarisen ohjelmoinnin ongelmia voidaan ratkaista visuaalisesti. Monimutkaisempia lineaarisen ohjelmoinnin ongelmia voidaan ratkaista Simplex-menetelmällä, joka on lineaarisen ohjelmoinnin ratkaisumenetelmä. Simplex-menetelmässä ongelmaa käsitellään sen perusmuodossa, ja Simplex-algoritmi etenee vaihe vaiheelta, kunnes se päätyy sellaiseen tilaan, että prosessi pysähtyy ja ongelmaan saadaan optimoitu vastaus.

Lineaarisen ohjelmoinnin opetuksesta toisella asteella on olemassa tutkimuksia, joita on tehty muun muassa Indonesiassa. Näistä tutkimuksista käy ilmi, että aiemman opitun tiedon omaksuminen on tärkeää lineaarisen ohjelmoinnin oppimisessa. On myös tukittu opetussarjakuvien ja opetusvideoiden käyttöä lineaarisen ohjelmoinnin opetuksessa. Tutkimusten perusteella ne tukevat oppilaiden lineaarisen ohjelmoinnin oppimista.

Suomen lukiossa pitkän matematiikan kurssit sisältävät oleelliset lähtötiedot, että opiskelija voi aloittaa perehtymisen lineaarisen ohjelmoinnin perusteisiin. Lineaarisen ohjelmoinnin ongelmat vaativat ongelmanratkaisukykyä ja sen ongelmien kautta opiskelijat voidaan perehdyttää erilaisten sähköisten ohjelmien käyttöön matematiikassa. Yksi tapa opettaa lineaarista ohjelmointia voisi olla projektimuotoinen opetus, jossa opiskelijoilla on yksi isompi aihe, johon kurssin edetessä lisättäisiin asioita niin, että sen ratkaisuun tarvittaisiin aina askel askeleelta enemmän työkaluja.

Avainsanat – Nyckelord – Keywords
lukio-opetus, lineaarinen ohjelmointi

Säilytyspaikka – Förvaringställe – Where deposited
Kumpulan kampuskirjasto

Muita tietoja – Övriga uppgifter – Additional information

Sisältö

Luku 1

Johdanto

Lineaarinen ohjelmointi on matemaattista optimointia, jota voidaan hyödyntää monissa käytännön ongelmissa. Tässä tutkielmassa käydään läpi lineaarisen ohjelmoinnin ongelmia ja niiden ratkaisutapoja, sekä miten lineaarista ohjelmointia voisi esitellä lukio-opetuksessa.

Lineaarisen ohjelmoinnin ongelma muodostuu lineaarisista ehdoista ja objektifunktiosta. Lineaarisen ohjelmoinnin ongelma täyttää aksioomat, jotka ovat verrannollisuus, lisättävyys ja jatkuvuus. Ongelmat voivat olla yksinkertaisia, mutta usein lineaarisen ohjelmoinnin ongelmat muodostuvat useista ehdoista ja monista muuttujista. Tämän takia lineaarisen ohjelmoinnin ongelmilla on olemassa yleinen muoto, joka helpottaa ongelman käsittelyä eri ratkaisumenetelmillä. Yleisessä muodossa kaikki ongelman epäyhtälöt on muutettu perusmuotoon.

Lineaarisessa ohjelmoinnissa mallin rakentaminen ongelmasta on yhtä tärkeää kuin ongelman ratkaiseminen. Lineaarisen ohjelmoinnin malli voidaan rakentaa sarakemenetelmällä, jossa ongelman ehdot muodostetaan sen aktiviteeteistä. Tämä prosessi jatkuu järjestelmällisesti, kunnes kaikki ongelman tarpeelliset tiedot on muutettu sellaiseen muotoon, että ongelmaa voidaan käsitellä lineaarisen ohjelmoinnin työkaluilla

Lineaarisessa ohjelmoinnissa kaikki ehdot ja objektifunktio ovat lineaarisia, mikä helpottaa ongelman ratkaisualueen tutkimista. Tutkielmassa osoitetaan, että lineaarisessa ohjelmoinnissa ongelman suurin ja pienin arvo löytyvät ehtolauseiden rajaamien alueiden ääripisteistä. Tätä tietoa hyödyntämällä yksinkertaisia lineaarisen ohjelmoinnin ongelmia voidaan ratkaista visuaalisesti.

Kun yritetään ratkaista ongelmia, joita ei enää voida ratkaista visuaalisesti, pitää käyttää erilaisia ratkaisualgoritmeja. Yksi tällainen algoritmi on Simplex-algoritmi, jota käytetään Simplex-menetelmässä. Simplex-menetelmä on lineaarisen ohjelmoinnin ratkaisumenetelmä, jossa lineaarisen ohjelmoinnin ongelmaa käsitellään sen perusmuodossa, ja ratkaisu saadaan käyttämällä Simplex-algoritmia. Ongelmaa ratkaistaessa Simplex-algoritmi etenee vaihe vaiheelta, kunnes se päätyy sellaiseen tilaan, että prosessi pysähtyy ja ongelmaan saadaan optimoitu vastaus.

Erilaisia tapoja opettaa lineaarista ohjelmointia toisella asteella on tutkittu jonkin verran. Etenkin Indonesiassa on tehty näitä tutkimuksia. Tutkimuksista kävi ilmi, että aiemman opitun tiedon omaksuminen on tärkeää lineaarisen ohjelmoinnin asioiden oppimisessa, ja että opetusmetodeina opetussarjakuvat ja opetusvideot vaikuttavat tukevan oppilaiden oppimista.

Suomessa lukiokurssiksi lineaarinen ohjelmointi soveltuu sen takia, että pitkän matematiikan kurssit sisältävät kaikki oleelliset lähtötiedot, joiden avulla opiskelija voi aloittaa perehtymisen lineaarisen ohjelmoinnin perusteisiin. Lineaarisen ohjelmoinnin ongelmat vaativat ongelmanratkaisukykyä ja sen ongelmien kautta opiskelijat voidaan perehdyttää erilaisten sähköisten ohjelmien käyttöön matematiikassa. Yksi vaihtoehto opettaa lineaarista ohjelmointia olisi projektimuotoinen opetus, jossa opiskelijoilla olisi yksi isompi aihe, johon kurssin edetessä lisättäisiin asioita, niin että sen ratkaisuun tarvittaisiin aina askel askeleelta enemmän työkaluja.

Työssä esiintyviin esimerkkitehtäviin, jotka eivät ole itse tehtyjä, on selvästi merkitty lähteet.

Luku 2

Lineaarisen ohjelmoinnin ongelmat

Lineaarisesta ohjelmoinnista löytyy monia sovelluksia eri alojen ongelmien ratkaisuun. Käytännön ongelmissa ei kuitenkaan aina ole selvää, mitä pitäisi laskea. Ennen kuin ongelmia voidaan ratkaista, pitää ymmärtää, mikä ongelma oikeasti on. Kaikilla lineaarisen ohjelmoinnin ongelmilla on yhteisiä piirteitä, ja on olemassa erilaisia käytäntöjä, joiden avulla voidaan muodostaa erilaisista ongelmista lineaarisessa ohjelmoinnissa laskettavia malleja. [2, s. 1-2]

Katsotaan, miltä yksinkertainen lineaarisen ohjelmoinnin ongelma näyttää ja miten sen voisi ratkaista.

Lukiolaisella on 100 minuuttia aikaa tehdä kotitehtäviä. Hän voi tehdä äidinkielen tai matematiikan tehtäviä. Opiskelija saa jokaisesta matematiikan tehtävästä 2 lisäpistettä tulevaan matematiikan kokeeseen ja 5 lisäpistettä jokaisesta tehdystä äidinkielen tehtävästä tulevaan äidinkielen kokeeseen. Aikaa häneltä kuluu 5 minuuttia yhden matematiikan tehtävän tekemiseen ja 15 minuuttia yhden äidinkielen tehtävän tekemiseen. Vajaista tehtävistä saa pisteitä siinä suhteessa, kuinka paljon tarvittavasta ajasta niiden tekemiseen on käytetty. Miten opiskelijan kannattaa tehdä tehtäviä, että hän saa mahdollisimman monta lisäpistettä tulevan koeviikon kokeisiin?

Valitaan muuttujiksi matematiikan tehtävät, joita merkitään symbolilla x_1, ja äidinkielen tehtävät, joita merkitään symbolilla x_2. Huomataan, että aika rajoittaa, kuinka paljon tehtäviä voidaan tehdä. Koska äidinkielen tehtävän tekoon menee 15 minuuttia ja matematiikan tehtäviin 5 minuuttia, voidaan muodostaa epäyhtälö

$$5x_1 + 15x_2 \leq 100$$

Seuraavaksi huomataan, että opiskelijan lisäpisteet muodostuvat seuraavasti

$$2x_1 + 5x_2 = z$$

Yhtälössä z kertoo kokonaispisteiden määrän.

Ongelman vastaus voidaan päätellä, kun tiedetään, että äidinkielen tehtävään menee aikaa 3 kertaa enemmän, mutta siitä saadaan vain 2,5 kertaa enemmän pisteitä. Tässä tapauksessa parhaat pisteet voi kerätä tekemällä pelkästään matematiikan tehtäviä, jolloin ainoastaan aikaraja vaikuttaa pisteisiin. Asetetaan äidinkielen tehtäville arvo 0 ja ratkaistaan ajankäytön epäyhtälö:

$$5x_1 \leq 100| : 5$$

$$x_1 \leq 20$$

Eli voidaan tehdä enintään 20 tehtävää, joista saadaan pisteitä $2 \cdot 20 = 40$.

Ongelmaan löytyy vastaus varsin helposti, mutta mitä jos kyseessä olisikin opintopiiri, jonka yhteispisteet pitäisi maksimoida. Ongelmaa vaikeuttaisi myös se, jos opintopiirin eri opiskelijat tekisivät eri aineiden tehtäviä eri nopeuksilla. Mitä jos tilanteessa pitäisi vielä ottaa kaikki lukiossa opetettavat aineet huomioon? Lineaarisessa ohjelmoinnissa on onneksi paljon työkaluja ja menetelmiä, joiden avulla monimutkaisempiakin ongelmia voidaan käsitellä selkeästi.

2.1 Lineaarisen ohjelmoinnin ongelmien rakenne

Lineaarisen ohjelmoinnin ongelmat muodostuvat objektifunktiosta ja yhdestä tai useammasta lineaarisesta ehdosta. Ratkaistaessa lineaarisen ohjelmoinnin ongelmaa tulee objektifunktiolle löytää paras mahdollinen ratkaisu niin, että kaikki ehtolauseet pysyvät paikkansapitävinä. Ongelman ehdot muodostuvat sen eri osista, ja ne riippuvat ongelman luonteesta. Tyypillisiä lineaarisen ohjelmoinnin ongelmia ovat muun muassa kuljetusongelma, metallien sekoitusongelma ja työntekijöiden ajankäytön optimointiongelma. [2, s. 33-34]

Kuljetusongelmassa kuljetetaan tuotteita paikasta toiseen. Sen ehdot rakentuvat tavaran liikkumisesta ja kuljetuksen kustannuksista. Annettujen ehtojen perusteella muodostetaan lopulta ratkaisu, jossa kustannukset ovat mahdollisimman pienet ja kaikki annetut ehdot toteutuvat. [2, s. 2-4]

Metallien sekoitus -ongelmassa yritetään saavuttaa haluttu seosmetalli, jossa eri metallien pitoisuudet ovat oikeissa suhteissa keskenään. Sekoitusprosessissa ei käytetä aina puhtaita metalleja, vaan haluttu metalliseos on seosmetallien yhdistelmä. Kun ongelmaa lähdetään rakentamaan, ehdot muodostuvat eri seosten vaihtoehdoista ja niiden hinnoista. Lopputuloksena on mahdollisimman taloudellinen tapa valmistaa haluttua seosmetallia. [3, s. 3]

Lineaarinen ohjelmointi soveltuu myös työpaikan työntekijöiden ajankäytön optimointiin. Ongelman muoto voi olla sellainen, että työpaikalla on tietty määrä ihmisiä, joiden aika jakautuu työnteon ja uusien työntekijöiden koulutuksen kesken. Nyt ongelman ehdot rakennetaan siten, että pystytään tasapainottamaan tarpeellisten työntekijöiden koulutus ja vaadittu tuotantotaso niin, että työntekijöitä on juuri oikea määrä ja kaikki sovitut työt tulevat tehdyiksi. [2, s. 4-5]

2.2 Lineaarisen ohjelmoinnin ongelmien aksioomat

Lineaarisen ohjelmoinnin ongelmien määritelmä sisältää tiettyjä perusoletuksia ongelman luonteen suhteen. Nämä aksioomat ovat verrannollisuus, lisättävyys ja jatkuvuus. Muun tyyppiset matemaattisen ohjelmoinnin ongelmat eivät täytä näitä kaikkia aksioomia. Esimerkiksi kokonaislukuohjelmoinnin ongelmissa jatkuvuuden ehto ei täyty. [3, s. 22]

2.2.1 Verrannollisuus

Kun yhdestä leivästä saatavien kalorien määrä on 100,5 kaloria, silloin kahdesta leivästä saatavien kalorien määrä on 201,0 kaloria. Verrannollisuudella lineaarisessa ohjelmoinnissa tarkoitetaan sitä, että kaikki suhteet pysyvät vakioina. [3, s. 22]

2.2.2 Lisättävyys

Lisättävyys tarkoittaa lineaarisessa ohjelmoinnissa sitä, että jos ongelmassa halutaan lisätä tuotantoa, silloin pitää myös lisätä resurssien määrää samassa suhteessa. Oletetaan, että leivästä saadaan 100,5 kaloria ja kananmunasta saadaan 50 kaloria. Silloin aamiaisella, jossa on 2 leipää ja kananmuna, on yhteensä $100,5 \cdot 2 + 50 = 251$ kaloria. [3, s. 22-23]

2.2.3 Jatkuvuus

Ongelman muuttujat ovat jatkuvia, eli ongelman muuttujat voivat saada minkä tahansa reaaliarvon ongelman ehtojen rajoittaman alueen sisällä. Esimerkiksi tilanteessa, jossa muuttuja vaatii pelkästään kokonaislukuja, tilanne ei ole jatkuva ja silloin se ei kuulu lineaarisen ohjelmoinnin ongelmiin. [3, s. 23]

2.3 Lineaarisen ohjelmoinnin ongelman yleismuoto

Lineaarisen ohjelmoinnin ongelmat muodostuvat lineaarisista ehdoista

$$a_1x_1 + a_2x_2 + ... + a_jx_j + ... + a_nx_n = b$$

Kertoimet a_j ja b ovat tunnettuja. Muuttujat x_j ovat tuntemattomia. Kun lineaarista ohjelmointia käsitellään tasossa, muoto on seuraava

$$a_1x_1 + a_2x_2 = b$$

Lineaaristen ehtojen lisäksi ongelmassa on objektifunktio, joka ilmaisee ongelman tavoitetta ja on riippuvainen lineaarisen ohjelmoinnin ongelman muuttujista. Objektifunktiossa arvoa, jota merkitään kirjaimella z, joko minimoidaan tai maksimoidaan. [4, s. 6]

Gass määrittelee lineaarisen ohjelmoinnin ongelman yleisen muodon seuraavasti kirjassaan [4, s. 6].

Määritelmä 2.3.1. *Lineaarisen ohjelmoinnin ongelman yleinen muoto on seuraava. Objektifunktio, jonka arvoa z minimoidaan tai maksimoidaan, on muotoa*

$$c_1x_1 + c_2x_2 + ... + c_jx_j + ... + c_nx_n = z,$$

jossa c_j ovat tunnettuja kertoimia, kun muuttujille x_j pätevät seuraavat rajoite-ehdot Gassin mukaan

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1j}x_j + ... + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + ... + a_{2j}x_j + ... + a_{2n}x_n = b_2$$
$$...$$
$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{ij}x_j + ... + a_{in}x_n = b_i$$
$$...$$
$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mj}x_j + ... + a_{mn}x_n = b_m$$
$$x_1, x_2, ..., x_j, ..., x_n \geq 0$$

Yhtälöryhmässä

$$j = 1, 2, ..., n$$
$$i = 1, 2, ..., m$$

ja a_{ij}:t ovat vakioita kaikille $n, m \in \mathbb{N}$ ja $b_i \geq 0$.

Jos yhtälössä b_i saa arvon 0, lineaarista ohjelmoinnin ongelmaa kutsutaan degeneroituneeksi ja sillä voi olla useita ratkaisuja. [2, s. 81]

2.4 Lineaarisen ohjelmoinnin epäyhtälöt

Usein lineaarisen ohjelmoinnin ongelmat muodostuvat epäyhtälöistä, mutta lineaarisen ohjelmoinnin perusmuodossa niitä ei ole. Tätä varten epäyhtälöt pitää muuttaa perusmuotoon lisäämällä niihin tarvittavat ylijäämämuuttujat. Esimerkiksi epäyhtälöön

$$x_1 + ... + x_n \leq b$$

lisäämällä tarvittava ylijäämämuuttuja $x_{n+1} \geq 0$ päästään perusmuotoon

$$x_1 + ... + x_n + x_{n+1} = b$$

[2, s. 86]. Jos käsiteltävänä on useampi epäyhtälö, niin jokaiseen epäyhtälöön pitää lisätä erillinen ylijäämämuuttuja.

Jos epäyhtälö on "suurempi tai yhtä suuri kuin" -muotoa, ylijäämämuuttuja pitää vähentää yhtälöstä, ja jos se taas on "pienempi kuin tai yhtä suuri kuin" -muotoa, ylijäämämuuttuja pitää lisätä yhtälöön.

Epäyhtälöiden perusmuotoon muuttaminen tapahtuu seuraavasti. Otetaan esimerkiksi seuraava yhtälöpari käsiteltäväksi.

$$3x_1 + x_2 \geq 8$$

$$2x_1 + 4x_2 \geq 3$$

Lisätään ylijäämämuuttujat $x_3 \geq 0$ ja $x_4 \geq 0$

$$3x_1 + x_2 - x_3 = 8$$

$$2x_1 + 4x_2 - x_4 = 3$$

Näin ollaan päästy yhtälöiden perusmuotoihin.

Luku 3

Lineaarisen ohjelmoinnin mallien rakentaminen

Lineaarisessa ohjelmoinnissa ongelman muodostaminen on yhtä merkittävä osa ratkaisuprosessia kuin numeerisessa muodossa olevan ongelman ratkaiseminen[3, s. 8]. Ongelmasta pitää poimia tärkeät tiedot, joiden pohjalta rakennetaan ongelmalle malli. Tämän mallin lopputulos puolestaan voidaan ratkaista erilaisilla lineaarisen ohjelmoinnin työkaluilla. Joskus ongelmaa joudutaan käymään useaan kertaan lävitse, jotta saadaan kerättyä kaikki tarpeellinen tieto [2, s. 35]. Lopullisessa mallissa tulee olla kaikki ongelman kannalta merkittävät relaatiot.

3.1 Mallin rakentaminen sarakemenetelmällä

Yksi mallin rakennustavoista on sarakemenetelmä. Sarakemenetelmää kutsutaan myös aktiviteettimenetelmäksi, koska siinä muodostetaan ongelman ehdot aktiviteeteista. Ideana on se, että on olemassa musta laatikko, jonka sisällä tapahtuu jokin aktiviteetti. Mustalle laatikolle annetaan syöte, ja se antaa takaisin tulosteen. Tämä tarkoittaa, että kun ongelmassa on resursseja, jotka toimivat syötteenä, niiden avulla saadaan tietty määrä haluttuja tuotteita. [3, s. 9]

Lineaarisessa ohjelmoinnissa ongelmien suhde on puhtaasti lineaarinen. Ainoa tapa kasvattaa tuotantoa on kasvattaa resurssien määrää. Tämän takia täsmällisen mallin rakentaminen on tärkeää, että siitä voidaan löytää ongelman pullonkaulat, jolloin tuotanto pystytään optimoimaan.

Sarakemenetelmässä ongelman mallinnus tapahtuu seuraavissa vaiheissa.

Vaihe 1: Määritellään aktiviteetit muuttujiksi. Muuttuja annetaan kaikille asioille, joita pystytään ongelman prosessissa mittaamaan. Nämä muuttujat nimetään juoksevalla indeksillä niin, että saadaan lopulta muuttujajoukko $x_1, x_2, ..., x_n$, jossa indeksiluvun perusteella tiedetään aktiviteetin sijainti. Ongelman aktiviteetteja kuvataan yleensä indeksin arvolla j.

Vaihe 2: Määritellään tuotteet. Ongelman prosessissa aktiviteetit kuluttavat ja tuottavat tuotteita. Ongelman kannalta on oleellista valita jokin yhteinen mittausyksikkö kaikille tuotteille. Tämä mittayksikkö on usein sidoksissa kuluihin, jolloin se mittaa joko aikaa, resursseja tai rahaa. Ongelman tuotteita kuvataan indeksiarvolla i. **Vaihe 3:** Määritellään kertoimet aktiviteeteille. Koska aktiviteetit kuluttavat ja tuottavat tuotteita, tulee prosessin vaiheille löytää kertoimet, jotka kuvaavat sitä, missä suhteessa kulutettavat resurssit tuottavat haluttuja tuotteita. Tämä kerroin esitetään perusmuodossa merkinällä "a_{ij}", jossa indeksi i viittaa tuotteeseen ja j aktiviteettiin.

Vaihe 4: Selvitetään prosessin ulkopuolelta tulevat ja ulkopuolelle menevät resurssit, jotka toimivat myös ongelman rajoittavina tekijöinä ja niitä kuvataan merkinnällä b_i.

Vaihe 5: Muodostetaan resurssien tasapainoyhtälöt. Annetaan muuttujat $x_1, x_2, ...$ kuvaamaan eri aktiviteettien määriä, jotka ovat ongelmassa tuntemattomia. Tämän jälkeen muodostetaan tasapainoyhtälöt kaikille tuotteille aktiviteettien suhteen. [3, s. 9-11]

3.2 Esimerkki mallin rakentamisesta

Tutkitaan seuraavaksi esimerkkiä, joka perustuu Danzigin [2, s. 35-42] esimerkkiin, kuinka kuljetusongelman tiedot muutetaan matemaattiseksi maliksi. Alkuperäisestä esimerkistä ainoastaan säilykkeet on muutettu maaliksi, auttamaan lineaaristen ohjelmoinnin ongelmien jatkuvuuden hahmottamista. Ongelma muodostuu seuraavista tiedoista

- Kaksi tehdasta, jotka tuottavat maalia
- Kolme varastoa, joilla on tarve tietylle määrälle maalia
- Ehtoja, jotka määrittävät, kuinka paljon maalia pitää kuljettaa ja miten kuljetuksesta muodostuu kuluja.

Numeroidaan tehtaat niin, että ongelmassa esiintyvät tehdas 1 ja tehdas 2. Merkitään varastoja puolestaan kirjaimilla. Jatkossa käytämme merkintöjä varasto A, varasto B ja varasto C.

Varastojen tarve maalille muodostuu seuraavasti

Maalin kysyntä	
300 kg	Varasto A
300 kg	Varasto B
300 kg	Varasto C
900 kg	**Yhteensä**

Tehtaat tuottavat maalia seuraavasti

Tehtaiden maalin tuotanto	
350 kg	Tehdas 1
650 kg	Tehdas 2
1000 kg	**Yhteensä**

Ylimääräiset 100 kg maalia varastoidaan tehtaiden omiin varastoihin.

Ongelmana on löytää ratkaisu, jossa kuljetuskustannukset pysyvät mahdollisimman alhaisina niin, että jokainen varasto kuitenkin saa tarpeellisen määrän maalia.

Maalin kuljetuksesta aiheutuvat kulut ovat seuraavat, kun kuljetetaan 1 kg maalia paikasta toiseen

	Varastot		
Tehtaat	Varasto A	Varasto B	Varasto C
Tehdas 1	2,5	1,7	1,8
Tehdas 2	2,5	1,8	1,4

Seuraavaksi ongelma pitää muuttaa lineaarisen ohjelmoinnin malliin. Tämän mallin avulla se voidaan ratkaista lineaarisen ohjelmoinnin työkaluilla.

Vaihe 1: Aloitetaan hajottamalla ongelma pienempiin kokonaisuuksiin. Muodostetaan ensin kaikki 8 eri vaihtoehtoa sille, miten maalia siirtyy tehtaista säilytykseen.

Numero	Kulku
1	$1 \to A$
2	$1 \to B$
3	$1 \to C$
4	$2 \to A$
5	$2 \to B$
6	$2 \to C$
7	$1 \to 1$
8	$2 \to 2$

Vaihe 2: Seuraavaksi käydään läpi tuotteet, joita valmistetaan tai kulutetaan. Kustannukset lasketaan kuljetuskuluihin valitun mittayksikön avulla. Koska ongelmassa esiintyy vain yksi tuote, kaikki kustannukset on sidottu sen kuljetukseen.

Vaihe 3: Luodaan taulukko, josta näkyy, miten tuotteet kulkevat paikasta toiseen. Jos tehtaalta lähtee yksi kilogramma maalia ja se saapuu varastoon, taulukkoon pitää merkitä vastaava muutos. Tämän lisäksi merkitään, kuinka paljon kuluja kyseisestä muutoksesta seuraa.

Aktiviteetti	1	2	3	4	5	6	7	8
Tuotteet	$1 \to A$	$1 \to B$	$1 \to C$	$2 \to A$	$2 \to B$	$2 \to C$	$1 \to 1$	$2 \to 2$
Tehdas 1	$+1$	$+1$	$+1$				$+1$	
Tehdas 2				$+1$	$+1$	$+1$		$+1$
Varastossa A	-1			-1				
Varastossa B		-1			-1			
Varastossa C			-1			-1		
Kulut	$+2,5$	$+1,7$	$+1,8$	$+2,5$	$+1,8$	$+1,4$		

Vaihe 4: Seurataan tavaran liikettä ongelmassa sekä sitä, mitä rajoituksia liikkumiselle on. Jokainen varasto tarvitsee 300 kg maalia, tehdas 1 tuottaa 350 kg maalia ja tehdas 2 tuottaa 650 kg maalia. Koska maalia tuotetaan 1000 kg mutta sitä tarvitaan vain 900 kg varastoihin, ylimääräiset 100 kg maalia jäävät tehtaiden omiin varastoihin.

Merkitään kaikkia kustannuksia symbolilla z. Ongelman kustannukset muodostuvat tuotteiden kuljetuksista, joiden ehdot muodostuvat tuotannon ja varaston asettamista ehdoista. Ongelmassa pyritään muodostamaan ratkaisu, jossa kulut eli z saa mahdollisimman pienen arvon. Kuvataan ongelman ehtoja ja kustannuksia seuraavalla taulukolla.

Tuotteet	Tavaran kulku	
1. Tehdas 1	350	Tarjoilla oleva tuotanto
2. Tehdas 2	650	
3. Varastossa A	-300	Täytettävä kysyntä
4. Varastossa B	-300	
5. Varastossa C	-300	
6. Kustannukset	z	Minimoitavat kustannukset

Vaihe 5: Määritellään seuraavaksi tasapainoyhtälöt. Otetaan vaiheen 1 taulukosta numeroidut muuttujat. Jokaiselle kilon siirtymälle on annettu muuttuja x_i, jossa $i \in [1, ..., 8]$. Kun nämä muuttujat yhdistetään kohdan kolme taulukon tietojen kanssa, voidaan muodostaa tasapainolausekkeet tavaran määrille. Ulos menevien kilojen kaavat muodostuvat seuraavasti

$$1 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3 + 1 \cdot x_7$$

Nyt myös tiedetään, että tehdas 1 tuottaa 350 kg maalia, ja tästä seuraa

$$x_1 + x_2 + x_3 + x_7 = 350.$$

Samaa periaatetta voidaan myös soveltaa tehtaalle 2

$$x_4 + x_5 + x_6 + x_8 = 650.$$

Seuraavaksi muodostetaan varastojen kaavat. Niissä hyödynnetään myös vaiheen 3 tietoja. Saadaan seuraava kaava varastolle A

$$-x_1 - x_4 = -300$$

varastolle B

$$-x_2 - x_5 = -300$$

ja varastolle C

$$-x_3 - x_6 = -300.$$

Tehtaiden ja varastojen kaavojen jälkeen pitää muodostaa kululaskelma seuraavasti.

$$2,5x_1 + 1,7x_2 + 1,8x_3 + 2,5x_4 + 1,8x_5 + 1,4x_6$$

Lisätään kaavaan vielä muuttuja z kuvaamaan kokonaiskuluja, jolloin päästään muotoon

$$2,5x_1 + 1,7x_2 + 1,8x_3 + 2,5x_4 + 1,8x_5 + 1,4x_6 = z$$

Kun kaikki tarvittava tieto on kerätty, voidaan muodostaa lineaarisen ohjelmoinnin ongelman yleinen muoto. Objektifunktio on

$$2,5x_1 + 1,7x_2 + 1,8x_3 + 2,5x_4 + 1,8x_5 + 1,4x_6 = z$$

ja siihen pätevät seuraavat ehdot:

$$x_1 + x_2 + x_3 + x_7 = 350$$

$$x_4 + x_5 + x_6 + x_8 = 650$$

$$x_1 + x_4 = 300$$

$$x_2 + x_5 = 300$$

$$x_3 + x_6 = 300$$

Luku 4

Yksinkertaisten lineaarisen ohjelmoinnin ongelmien ratkaiseminen

Lineaarisen ohjelmoinnin ongelmia voidaan ratkaista monella tapaa. Niin kauan kun ongelma pysyy yksinkertaisena, on valittavana myös monia toimivia ratkaisutapoja. Lineaarisen ohjelmoinnin ongelmia voidaan ratkaista puhtaasti visuaalisesti niin pitkään, kun ongelma pysyy tasossa [3, s. 35].

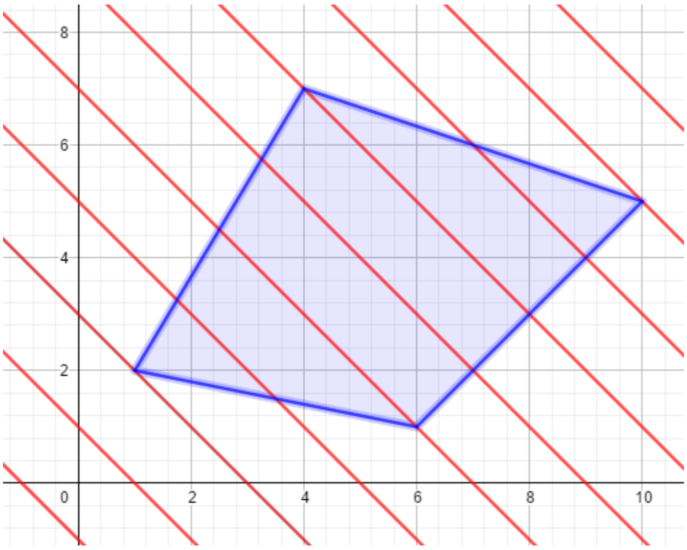
Jotta lineaarisen ohjelmoinnin ongelmat voidaan ratkaista tasossa visuaalisesti, tarvitaan seuraava lause:

Lause 4.0.1. *Lineaarisen ohjelmoinnin ongelman suurin ja pienin arvo löytyvät rajatun alueen ääripisteistä, kun ollaan tasossa.*

Todistus. Oletetaan, että meillä on lineaarisista yhtälöistä muodostettu rajattu alue A ja objektifunktio $z = c_1x_1 + c_2x_2$. Koska objektifunktio on lineaarinen, huomataan, että sen osittaisderivaatat $\frac{\partial z}{\partial x_1}$ ja $\frac{\partial z}{\partial x_2}$ saavat arvon 0 vain, jos niiden kerroin on 0. Tästä seuraa se, että objektifunktiolla ei ole lokaaleja nollakohtia, vaan sen suurin ja pienin arvo löytyvät sen päädyistä. Objektifunktion ääriarvot löytyvät siitä kohdasta, missä se leikkaa alueen A reunan.

Nyt tiedetään, että objektifunktio saa ääriarvonsa alueen A reunalla. Tiedetään myös, että ongelman A reuna muodostuu lineaarisista ehdoista. Jos objektifunktio leikkaa alueen A kohdassa u, joka ei ole kärkipiste, ja saa siinä myös suurimman arvonsa, huomataan reunan ollessa lineaarinen, että sillä liikuttaessa objektifunktion arvo joko kasvaa, pienenee tai pysyy samana. Jos reunapisteellä liikuttaessa objektifunktion arvo pienenee, sitä voidaan kasvattaa liikkumalla toiseen suuntaan. Nyt kuitenkin huomataan, että jos ollaan kahden kärkipisteen välissä, suurin arvo ei voi pienentyä, koska muuten liikkumalla toiseen suuntaan sen pitäisi kasvaa. □

Visuaalisesti lausetta 4.0.1 voi havainnollistaa piirtämällä rajoitetun alueen, jossa pitää maksimoida objektifunktiota $c_1x_1 + c_2x_2 = z$. Sijoittamalla suoria eri z arvoilla saadaan seuraava kuva eräillä c_1 ja c_2 arvoilla.



Kuvasta käy selkeästi ilmi, kun käydään läpi mahdollisia z:n arvoja, että se saa suurimman ja pienimmän arvonsa rajoitetun alueen kärkipisteissä.

Lause 4.0.2. *Avaruudessa $\mathbb{R}^n$ lineaarisen ohjelmoinnin ongelman ratkaisut löytyvät alueen ääripisteistä.*

Todistus. Huomataan, samoin kuin tasossa, että myöskään euklidisessa avaruudessa lineaarisella funktion derivaatalla ei ole lokaaleja nollakohtia, ellei kyseessä on nolla-funktio. Olkoon objektifunktio $z = c_1x_1 + ... + c_ix_i + ...c_nx_n$, jossa $1 \leq i \leq n$. Kyseisen objektifunktion osittaisderivaatalla $\frac{\partial z}{\partial x_i} = 0$ ei ole nollakohtaa, jos $c_i \neq 0$. Tästä seuraa se, että ääriarvo löytyy myös euklidisessa avaruudessa lineaarisen ongelman rajoitetun alueen reunoilta.

Lineaarisista ehdoista muodostuu alue A. Voidaan valita alueelta mitkä tahansa kolme kärkipistettä ja virittää niiden avulla taso. Kun tutkitaan tasolla objektifunktion ääripisteitä, voidaan hyödyntää aikaisempaa todistusta 4.0.1, jonka perustella tiedetään, että tasossa objektifunktion suurin ja pienin arvo löytyvät rajoitetun alueen kärkipisteistä. Ongelman suurin ja pienin arvo löytyvät siis alueen $A \in \mathbb{R}^n$ kärkipisteistä. $\square$

4.1 Lineaarisen ohjelmoinnin ongelman ratkaiseminen visuaalisesti

Yksinkertaiset lineaarisen ohjelmoinnin ongelmat voidaan ratkaista visuaalisesti. Kun lineaarisen ohjelmoinnin ongelmassa on vain kaksi positiivista muuttujaa, silloin ongelma voidaan ratkaista piirtämällä kaikki ongelman ehdot koordinaattiakselille ja rajoitetun alueen kärkipisteistä saadaan ongelman suurin ja pienin arvo. [3, s. 35]

Tutkitaan esimerkkiä Danzigin kirjasta [3, s. 35-36], jossa pitää minimoida objektifunktio

$$-2x_1 - x_2 = z$$

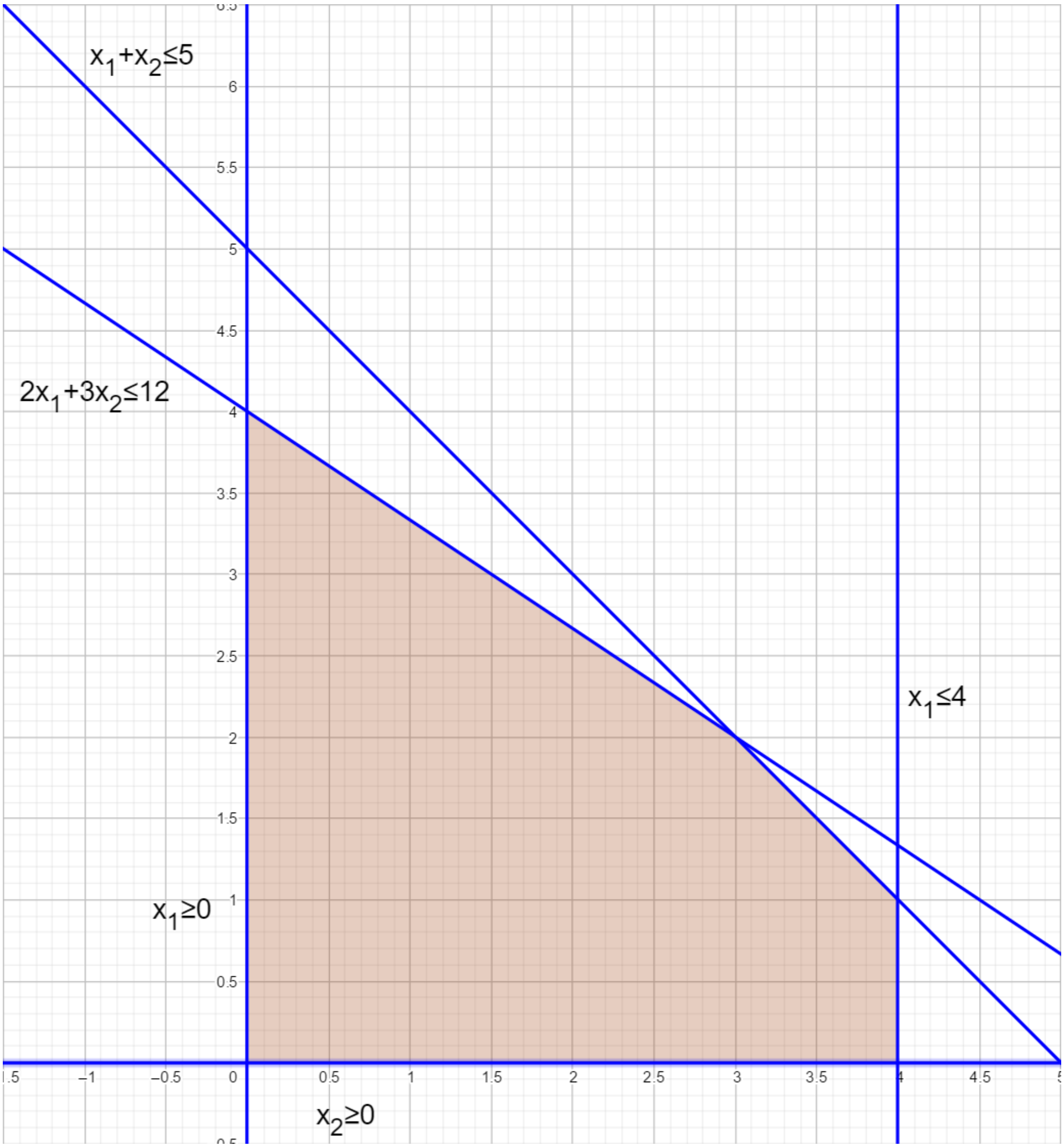
seuraavien ehtojen rajoittamassa alueessa

$$x_1 + x_2 \leq 5$$

$$2x_1 + 3x_2 \leq 12$$

$$x_1 \leq 4$$

kun $x_1 \geq 0, x_2 \geq 0$.



Ongelman kuvasta voidaan etsiä leikkauskohdat, tai ne voidaan löytää ratkaisemalla ongelman yhtälöparit keskenään. Alueen kärkipisteiksi saadaan $(0, 0)$, $(4, 0)$, $(0, 4)$, $(4, 1)$ ja $(3, 2)$. Sijoitetaan nämä saadut koordinaatit objektifunktioon.

$$-2 \cdot 0 - 0 = 0$$

$$-2 \cdot 4 - 0 = -8$$

$$-2 \cdot 4 - 1 = -9$$
$$-2 \cdot 0 - 4 = -4$$
$$-2 \cdot 3 - 2 = -7$$

Objektifunktio saa pienimmän arvonsa pisteessä $(4, 1)$.

4.1.1 Esimerkki sanallisen tehtävän visuaalisesta ratkaisemisesta

Palataan hetkeksi tämän työn alussa olleen esimerkin teemaan, jossa opiskelija yritti valita parhaan tavan tehdä tehtäviä, jotta hän saisi maksimimäärän lisäpisteitä. Tutkitaan, miten ongelman voisi ratkaista, jos ongelmassa olisikin kyse kahden hengen opintopiiristä, jossa on kaksi eri tason osaajaa tekemässä yhdessä tehtäviä.

Ongelma: Kaksi opiskelijaa tekee yhdessä tehtäviä. Opiskelijalla 1 on aikaa 90 minuuttia ja opiskelijalla 2 vain 60 minuuttia. Opiskelijalta 1 menee aikaa matematiikan tehtäviin 10 minuuttia ja äidinkielen tehtäviin 15 minuuttia. Opiskelijalta 2 puolestaan menee yhteen matematiikan tehtävään 15 minuuttia, mutta äidinkielen tehtävään 5 minuuttia. Yhdestä matematiikan tehtävästä saa 5 lisäpistettä ja yhdestä äidinkielen tehtävästä 10 lisäpistettä tulevalle koeviikolle. Miten opiskelijoiden kannattaa tehdä tehtäviä yhdessä, jotta he saavat mahdollisimman monta pistettä?

Aloitetaan muodostamaan ratkaisua. Ehdot muodostetaan opiskelijoiden tehtäviin kuluvasta ajasta, ja objektifunktio muodostuu saatavissa olevista pisteistä seuraavasti

$$10x_1 + 15x_2 \leq 90$$
$$15x_1 + 5x_2 \leq 60$$
$$5x_1 + 10x_2 = z$$

kun $x_1 \geq 0$, $x_2 \geq 0$

Piirretään tilanteesta kuva.

Alueen kärkipisteiksi saadaan $(0,0),(4,0),(0,6)$, ja $(\frac{18}{7},\frac{30}{7}) \approx (2.6, 4.3)$. Viimeiseksi kärkipisteeksi voidaan valita likiarvo kuvan perusteella tai laskea tarkka arvo ratkaisemalla yhtälöpari $10x_1 + 15x_2 = 90$ ja $15x_1 + 5x_2 = 60$. Sijoitetaan nämä koordinaatit objektifunktioon.

$$5 \cdot 0 + 10 \cdot 0 = 0$$

$$5 \cdot 4 + 10 \cdot 0 = 20$$

$$5 \cdot 0 + 10 \cdot 6 = 60$$

$$5 \cdot 2,6 + 10 \cdot 4,3 = 56$$

Opintopiiri pääsee siis parhaaseen tulokseen tekemällä pelkästään äidinkielen tehtäviä.

Luku 5

Simplex-menetelmä

Kun käsitellään monimutkaisia lineaarisen ohjelmoinnin ongelmia, niitä on järkevää käsitellä erilaisten algoritmien avulla. Yksi tällainen algoritmi on Simplex-algoritmi, jota sovelletaan Simplex-menetelmässä. Vaikka teoreettisella tasolla Simplex-menetelmä voi olla haastava käsite, mekaanisesti laskettaessa se toimii yksinkertaisesti.

Simplex-menetelmä etenee kahdessa vaiheessa. Ensimmäisessä vaiheessa käsittelyssä on lineaarisen ohjelmoinnin ongelma perusmuodossa, ja se muutetaan kanoniseen muotoon. Toisessa vaiheessa sovelletaan ongelman kanoniseen muotoon Simplex-algoritmia, jonka avulla ongelma ratkaistaan. [2, s. 94]

Simplex-menetelmä vaatii sitä, että ongelman kaikki ehtolauseet on muutettu epäyhtälöistä yhtälöiksi lisäämällä niihin tarvittavat ylijäämämuuttujat. Esimerkiksi epäyhtälöön

$$x_1a_1 + ... + x_na_n \leq b$$

lisätään apumuuttuja $x_{n+1} \geq 0$ ja päästään muotoon

$$x_1a_1 + ... + x_na_n + x_{n+1} = b.$$

[2, s. 86]

5.1 Lineaarisen ohjelmoinnin ongelman kanoninen muoto

Kanonisella muodolla tarkoitetaan muotoa, jossa kaikki kaavat ovat lineaarisesti riippuvaisia. Kanoninen muoto muodostuu muuttujista, joita kutsutaan kantamuuttujiksi. Kantamuuttujilla tarkoitetaan niitä muuttujia, jotka esiintyvät ainoastaan yhdessä ehtoyhtälössä. Kanonisen muodon yleinen esitysmuoto on

$$x_1 + \bar{a}_{1,m+1}x_{m+1} + ... + \bar{a}_{1j}x_j + ... + \bar{a}_{1n}x_n = \bar{b}_1$$
$$x_2 + \bar{a}_{2,m+1}x_{m+1} + ... + \bar{a}_{2j}x_j + ... + \bar{a}_{2n}x_n = \bar{b}_2$$
$$...$$
$$x_m + \bar{a}_{m,m+1}x_{m+1} + ... + \bar{a}_{mj}x_j + ... + \bar{a}_{mn}x_n = \bar{b}_m.$$

Jokaiselle x_i $(i \leq m)$ on olemassa vain yksi yhtälö, johon se sisältyy. [2, s. 73, 78]

Lineaarisen ohjelmoinnin yleisestä muodosta kanoniseen muotoon päästään alkeisoperaatioilla, jotka suoritetaan alkuperäiselle yhtälöjoukolle niin, että ongelman ratkaisut säilyvät paikkaansa pitävinä. Näitä alkeisoperaatioita on kaksi

1. Korvaa mikä tahansa yhtälö E_t yhtälöllä $[kE_t]$, missä $k \neq 0$.
2. Korvaa mikä tahansa yhtälö E_t yhtälöllä $[E_t + kE_i]$, missä E_i on mikä tahansa ongelman muu kuin yhtälö E_t.

[2, s. 73-74]

5.1.1 Esimerkki alkeisoperaatioiden käytöstä

Käyttämällä alkeisoperaatioita muutetaan seuraavat yhtälöt kanoniseen muotoon

$$x_1 - x_2 + x_3 = 2 \quad (E_1)$$
$$2x_1 + x_2 - x_3 = 7 \quad (E_2)$$

Määritelmän mukaan kantamuodossa jokainen kantamuuttuja esiintyy ainoastaan yhdessä yhtälössä, ja siitä yhtälöstä ei löydy muita kantamuuttujia. Valitaan kantamuuttujiksi x_1 ja x_2, ja tehdään tarvittavat alkeisoperaatiot. Korvataan yhtälö E_2 tekemällä sille seuraava alkeisoperaatio $E_2' = E_2 + E_1$, ja päästään tilanteeseen

$$x_1 - x_2 + x_3 = 2 \quad (E_1)$$
$$3x_1 = 9 \quad (E_2').$$

Korvataan E_1 tekemällä alkeisoperaatio $E_1' = E_1' - \frac{1}{3}E_2'$, jolloin päästään muotoon

$$-x_2 + x_3 = -2 \quad (E_1')$$
$$3x_1 = 9 \quad (E_2').$$

Korvataan vielä E_1' alkeisoperaatiolla $E_1'' = -1 \cdot E_1'$

$$x_2 - x_3 = 2 \quad (E_1'')$$
$$3x_1 = 9 \quad (E_2').$$

Nyt yhtälöt ovat kanonisessa muodossa, koska x_1 esiintyy vain yhtälössä E_2' ja x_2 esiintyy vain yhtälössä E_1''. [2, s. 74]

5.1.2 Simplex-algoritmi

Kun lineaarisen ohjelmoinnin ongelma on saatu kanoniseen muotoon, voidaan siihen soveltaa Simplex-algoritmia. Simplex-algoritmissa lineaarisen ohjelmoinnin ongelmia käsitellään taulukkomuodossa, jossa jokaiselle muuttujalle on varattu oma kohta ja yhtälöt esitetään omilla riveillään niin, että objektifunktio on alimmalla rivillä. Myöhemmin kohdassa 5.3 nähdään, miten tämä prosessi käytännössä toimii.

Yleisellä tasolla lineaarisen ohjelmoinnin ongelmaa käsitellään Simplex-algoritmissa seuraavasti, kun kyseessä on maksimointiongelma:

1. Valitaan objektifunktioriviltä pienin arvo c_i.
2. Tarkistetaan, onko arvo c_i negatiivinen. Jos arvo ei ole negatiivinen, lopetetaan prosessi.
3. Varmistetaan, että kantamuuttujien kertoimet ovat suurempia kuin 0 eli $I = \{i = 1, ..., m \mid \bar{a}_{is} > 0\} \neq \emptyset$. Jos tämä ei päde, pysähdytään.
4. Valitaan se rivi, josta löytyy arvo $p = \ min \frac{\bar{b}_i}{\bar{a}_{ir}}$. Kyseinen rivi on se rivi, jossa muuttuja indeksillä i on eniten riippuvainen ehdostaan.
5. Muutetaan p-muuttuja 1:ksi kertomalla rivi sen käänteisluvulla. Tämän jälkeen poistetaan muut arvot sarakkeesta, jossa p sijaitsee, alkeisoperaatioiden avulla.
6. Palataan kohtaan 1 ja toistetaan prosessi, kunnes joko kohdan 2 tai 3 ehto pysäyttää sen.

[5, s. 67]

Lineaarisessa ohjelmoinnissa Simplex-algoritmin vaiheessa 4 olevaa termiä p kutsutaan ohjelmoinnissa pivot-termiksi, ja vaiheen 5 alkeisoperaatioiden sarjaa kutsutaan puolestaan pivotoimiseksi.

Minimointiongelmassa ongelma pitää muuttaa maksimointitehtäväksi, joka tapahtuu taulukon matriisin transpoosilla, eli rivit muutetaan sarakkeiksi ja sarakkeet riveiksi.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

Saatu taulukko voidaan nyt ratkaista maksimointiongelman tapaan. Kohdassa 5.3.2 käydään läpi kaikki minimointiongelman ratkaisun vaiheet.

5.2 Simplex-menetelmän konvergenssi

Lineaarisen ohjelmoinnin ongelmia ratkaistaessa yritetään löytää objektifunktion z arvolle pienin tai suurin arvo, kun seuraavat ehdot pätevät

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2$$

$$...$$

$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{1n}x_n = b_m$$

$$c_1x_1 + c_2x_2 + ... + c_nx_n = z.$$

Simplex-algoritmin toimivuuden vuoksi tulee osoittaa, että se löytää lineaarisen ohjelmoinnin ongelman ratkaisut äärellisessä ajassa. Osoitetaan asia Dantzigin tekemän todistuksen [1, s. 1-6] avulla.

Lause 5.2.1. *Jos on olemassa optimimaalinen ratkaisu, on olemassa rajallinen määrä Simplex-algoritmin alkeisoperaatioita, joiden kautta päästään optimiratkaisuun, tai ratkaisuja on olemassa ääretön joukko, jolloin z:lle ei ole olemassa alarajaa.*

Ennen kun aloitetaan todistus, käydään läpi lauseen 5.2.1 kannalta oleellisia tietoja.

Otetaan lauseen 5.2.1 todistuksen tutkittavaksi lineaarisen ohjelmoinnin kanoninen muoto

$$x_1 + \bar{a}_{1m+1}x_{m+1} + ...\bar{a}_{1s}x_s + ...\bar{a}_{1n}x_n = \bar{b}_1$$

$$...$$

$$x_r + \bar{a}_{rm+1}x_{m+1} + ...\bar{a}_{rs}x_s + ...\bar{a}_{rn}x_n = \bar{b}_r$$

$$...$$

$$x_m + \bar{a}_{m,m+1}x_{m+1} + ...\bar{a}_{ms}x_s + ...\bar{a}_{mn}x_n = \bar{b}_m$$

$$\bar{c}_{m+1}x_{m+1} + ...\bar{c}_s x_s + ...\bar{c}_n x_n = z - \bar{z}_0.$$

Ongelmassa $\bar{z}_0$ on vakio ja $\bar{a}_{ij}$ ovat arvoja, jotka on saatu muodostettaessa yhtälöt niin, että muuttujat $x_1, ..., x_m$ löytyvät vain yhdestä yhtälöstä, kun $\bar{b}_i \geq 0$ kaikille $i = 1, 2, ..., m$.

Jos tilanne on degeneroitunut, niin pätee

$$\bar{b}_r = 0.$$

Koska arvo 0 ei muutu sitä kertomalla, yhtälöryhmä voi saada samoja ratkaisuja sille tehdyistä alkeisoperaatioista huolimatta. Tästä seuraa, että ongelmalle voidaan tehdä loputtomasti alkeisoperaatioita, jotka eivät johda mihinkään, jolloin ongelman ratkaiseminen ei onnistu tekemällä alkeisoperaatioita äärellisessä ajassa.

Osoitetaan todistuksen avuksi seuraava lemma.

Lemma 5.2.2. *Jos induktio-oletus pätee m-yhtälöryhmälle, missä $\bar{b}_i$ ei ole alkukantaratkaisussa arvoltaan 0, niin se pätee myös, kun $\bar{b}_i = 0$.*

Todistus. Muutetaan $\bar{b}_i = 0$ mihin tahansa positiiviseen muotoon, tässä tapauksessa annetaan sille arvo $\bar{b}'_i = 1$, ja nyt lemman mukaan on olemassa sellainen alkeisoperaatioiden sarja, että päästään haluttuun muotoon. Jos täsmälleen sama alkeisoperaatioiden sarja tehdään degeneroidulle ratkaisulle $\bar{b}'_i = 0$, niin kaikki alkukantaratkaisun ratkaisut säilyvät paikkaansa pitävinä. Koska haluttu ominaisuus viimeisessä kanonisessa muodossa riippuu vain kantamuuttujan valinnasta eikä yhtälön oikeasta puolesta, lemma on osoitettu todeksi. □

Voidaan aloittaa Simplex-menetelmän todistus.

Todistus. Todistetaan lause 5.2.1 induktiolla. Tehdään induktio-oletus. Oletetaan, että lause pätee, kun yhtälöiden määrä on $m-1$. Kun yhtälöiden määrä on pienempi kuin m, on olemassa äärellinen määrä alkeisoperaatioita, joiden avulla päästään kanoniseen muotoon. Silloin objektimuuttujan yhtälössä kaikki riippuvuudet ovat epänegatiivisia $\bar{c}_j \geq 0$ tai ongelmassa on olemassa sarake s, jossa $\bar{c}_s < 0$ ja kaikki sen riippuvaisuudet ovat positiivisia ($\bar{a}_{is} \geq 0$).

Perusaskel: Osoitetaan että tilanne pätee, kun yhtälöitä on vain yksi. Oletetaan, että alkukantaratkaisu ei ole degeneroitunut, eli $\bar{b}_1 > 0$. Nyt jokainen siitä saatu kantaratkaisu on myös epädegeneroitunut. Yhdelle yhtälölle voidaan tehdä tarvittava alkeisoperaatio, jotta päästään optimiratkaisuun. Tästä seuraa, että lause 5.2.1 pätee, kun yhtälöitä on yksi ja induktioehto pätee.

Induktioaskel: Oletetaan että väite pätee kaikille 1,2, ..., m-1 ja on olemassa ainakin yksi i, jolla $\bar{b}_i \neq 0$ m-yhtälöryhmässä. Jos emme ole optimaalisessa ratkaisussa, toistetaan alkeisoperaatioita, kunnes on päästy tilanteeseen, missä $\bar{z}_0$ ei voida enää pienentää degeneraation takia. Järjestetään yhtälöt uudelleen seuraavasti

$$\bar{b}_1 = \bar{b}_2 = ... = \bar{b}_r = 0$$

ja

$$\bar{b}_i \neq 0$$

kaikille $i = r + 1, ..., m$. Koska oletuksena on, että on olemassa $\bar{b}_1 \neq 0$, niin myös pätee, että $r < m$.

Induktio-oletuksen mukaan äärellisellä määrällä alkeisoperaatioita päästään yhtälöstä tilanteeseen, jossa kaikki $\bar{c}_j \geq 0$ tai jollekin s kaikki $\bar{a}_{is} \geq 0$, $1 \geq i \geq r$ ja $\bar{c}_s \geq 0$. Tilanteessa, jossa kantamuuttujat saavat arvon 0, niiden arvot myös pysyvät 0:ssa alkeisoperaatioiden jälkeenkin. Tämän seurauksena voimme tehdä koko yhtälöryhmälle samat alkeisoperaatiot ilman tarvetta korvata muuttujia $x_{r+1}, ..., x_m$.

Jos kantaratkaisulle, joka on kohdassa m, pätee $\bar{c}_j \geq 0$, koko yhtälöryhmällä on samat ominaisuudet. Vaihtoehtoisesti, jos lopullisessa kantaratkaisussa on s, niin $\bar{c}_s < 0$ ja $\bar{a}_{is} \geq 0$ kaikille $i = 1, 2, ...r$. Nyt joko $\bar{a}_{is} \geq 0$ kaikille $i = r + 1, ..., m$ tai muuttuja x_s voidaan lisätä yhtälösarjaan, jolloin saadaan positiivinen vähennys $\bar{z}_0$:ssa koska $\bar{b}_i > 0$ kaikille $i = r + 1, ..., m$. Koska z voi pienentyä vain rajallisen monta kertaa, alkeisoperaatioiden määrän pitää olla rajallinen, mutta se voi olla rajallinen vain, jos induktio-oletus pätee m-yhtälöryhmälle.

Nyt on osoitettu, että Simplex-algoritmi ratkaisee äärellisessä alkeisoperaatioiden määrässä yhtälösarjan, jossa yhtälöiden määrä on $m > 0$ ellei yhtälösarja ole degeneroitunut. Jos tilanne on degeneroitunut, niin lemmasta 5.2.2 seuraa, että tilanne ratkeaa edelleen äärellisessä alkeisoperaatioiden määrässä. [1, s. 1-6] □

5.3 Esimerkkejä Simplex-menetelmän käytöstä

5.3.1 Maksimointiesimerkki

Simplex-menetelmän käyttöä on hyvä käydä läpi esimerkkien avulla. Tähän työhön on valittu maksimointiesimerkiksi yksinkertainen kolmen muuttujan ja kahden ehtolauseen ongelma, koska siitä käy ilmi, miten Simplex-menetelmä toimii ilman, että ongelman läpikäymisessä olisi liian monia vaiheita.

Lineaarisessa ohjelmoinnin ongelmassa on annettu seuraavat tiedot. Objektifunktio, jonka arvo z tulee maksimoida, on

$$4x_1 + 6x_2 + 9x_3 = z \quad (x_1 \geq 0, x_2 \geq 0, x_3 \geq 0).$$

Yhtälöryhmälle pätevät seuraavat rajoitukset

$$\begin{aligned} 2x_1 + 5x_2 + 7x_3 &\leq 800 \\ 4x_1 + 3x_2 + 5x_3 &\leq 750. \end{aligned}$$

Lineaarisen ohjelmoinnin ongelman ratkaiseminen Simplex-menetelmällä alkaa järjestämällä yhtälöt niin, että kaikki muuttujat ovat samalla puolella, ja sen jälkeen yhtälöihin lisätään tarpeen mukaan ylijäämän muuttujat niin, että päästään seuraavaan muotoon

$$\begin{aligned} -4x_1 - 6x_2 - 9x_3 + z &= 0 \\ 2x_1 + 5x_2 + 7x_3 + x_4 &= 800 \quad (x_4 \geq 0) \\ 4x_1 + 3x_2 + 5x_3 + x_5 &= 750 \quad (x_5 \geq 0). \end{aligned}$$

Seuraava vaihe on muodostaa yhtälöiden tietojen perusteella taulukko. Nimetään taulukon rivit tunnuksilla R_1, R_2 ja R_3.

	x_1	x_2	x_3	x_4	x_5	z	
R_1	2	5	7	1	0	0	800
R_2	4	3	5	0	1	0	750
R_3	-4	-6	-9	0	0	1	0

Simplex-algoritmi jatkuu niin pitkään, kunnes alimman rivin, joka muodostuu objektifunktion arvoista, kaikki arvot ovat positiivisia. Aloitetaan Simplex-algoritmi valitsemalla negatiivisin arvo alimmalta riviltä, eli -9, ja tutkitaan sen sarakkeen arvoja.

	x_1	x_2	x_3	x_4	x_5	z	
R_1	2	5	7	1	0	0	800
R_2	4	3	5	0	1	0	750
R_3	-4	-6	-9	0	0	1	0

Valitaan arvo, joka on riippuvaisin rivin vakiosta. Tässä kohtaa se on 7, koska $\frac{800}{7} \leq \frac{750}{5}$. Seuraava vaihe on alkeisoperaatioita käyttämällä muuttaa valittu arvo 1:ksi. Tätä varten pitää tehdä seuraava alkeistoimitus $R_1' = R_1 \cdot \frac{1}{7}$.

	x_1	x_2	x_3	x_4	x_5	z	
R_1'	$\frac{2}{7}$	$\frac{5}{7}$	1	$\frac{1}{7}$	0	0	$\frac{800}{7}$
R_2	4	3	5	0	1	0	750
R_3	-4	-6	-9	0	0	1	0

Seuraavaksi muutetaan muita sarakkeen arvoja alkeisoperaatioilla niin, että ne saavat arvokseen 0. Tämä tehdään alkeisoperaatioilla $R_3' = R_3 + R_1' \cdot 9$ ja $R_2' = R_2 + R_1' \cdot (-5)$. Operaatioiden jälkeen päästään seuraavaan tilanteeseen.

	x_1	x_2	x_3	x_4	x_5	z	
R_1'	$\frac{2}{7}$	$\frac{5}{7}$	1	$\frac{1}{7}$	0	0	$\frac{800}{7}$
R_2'	$\frac{18}{7}$	-$\frac{4}{7}$	0	-$\frac{5}{7}$	1	0	$\frac{1250}{7}$
R_3'	-$\frac{10}{7}$	$\frac{3}{7}$	0	$\frac{9}{7}$	0	1	$\frac{7200}{7}$

Prosessia toistetaan, kunnes alimman rivin arvot ovat positiivisia. Jatketaan valitsemalla negatiivisin arvo $-\frac{10}{7}$ alimmalta riviltä.

	x_1	x_2	x_3	x_4	x_5	z	
R_1'	$\frac{2}{7}$	$\frac{5}{7}$	1	$\frac{1}{7}$	0	0	$\frac{800}{7}$
R_2'	$\frac{18}{7}$	-$\frac{4}{7}$	0	-$\frac{5}{7}$	1	0	$\frac{1250}{7}$
R_3'	-$\frac{10}{7}$	$\frac{3}{7}$	0	$\frac{9}{7}$	0	1	$\frac{7200}{7}$

Riippuvaisin arvo on nyt $\frac{18}{7}$. Tehdään alkeisoperaatio $R_2'' = R_2' \cdot \frac{7}{18}$, jonka jälkeen ollaan seuraavassa tilanteessa.

	x_1	x_2	x_3	x_4	x_5	z	
R_1'	$\frac{2}{7}$	$\frac{5}{7}$	1	$\frac{1}{7}$	0	0	$\frac{800}{7}$
R_2''	1	-$\frac{2}{9}$	0	-$\frac{5}{18}$	$\frac{7}{18}$	0	$\frac{625}{9}$
R_3'	-$\frac{10}{7}$	$\frac{3}{7}$	0	$\frac{9}{7}$	0	1	$\frac{7200}{7}$

Tehdään nyt alkeisoperaatiot $R_1'' = R_1' + R_2'' \cdot (-\frac{7}{2})$ ja $R_3'' = R_3' + R_2'' \cdot \frac{7}{10}$, joiden seurauksena saadaan sarakkeen muille muuttujille arvo 0.

	x_1	x_2	x_3	x_4	x_5	z	
R_1''	0	$\frac{7}{9}$	1	$\frac{2}{9}$	0	0	$\frac{850}{9}$
R_2''	1	-$\frac{2}{9}$	0	-$\frac{5}{18}$	$\frac{7}{18}$	0	$\frac{625}{9}$
R_3''	0	$\frac{1}{9}$	0	$\frac{8}{9}$	$\frac{5}{4}$	1	$\frac{10150}{9}$

Ollaan tilanteessa, jossa alimmalla rivillä kaikki kantamuuttujien arvot ovat positiivisia. Valitaan ne kantamuuttujat, joiden alimman rivin arvo jää nollaksi ja annetaan muiden kohtien muuttujille arvoksi 0. Nyt ollaan saatu ratkaisu ongelmalle.

$$x_1 = \frac{850}{9}$$

$$x_3 = \frac{625}{9}$$

$$x_3 = x_4 = x_5 = 0$$

Eli z:n maksimiarvo on

$$z = \frac{10150}{9}.$$

5.3.2 Minimointiesimerkki

Palataan taas kerran opintopiiriongelman pariin. Tällä kertaa muutetaan ongelmaa siten, että sen sijaan, että yrittäisimme saada mahdollisimman paljon lisäpisteitä tietyssä ajassa, yritämme saada tietyn pistemäärän mahdollisimman lyhyessä ajassa.

Valitaan kaksi opiskelijaa, jotka tekevät yhdessä tehtäviä opintopiirissä. Opiskelija 1 tarvitsee vähintään 60 lisäpistettä tulevalle koeviikolle ja opiskelija 2 tarvitsee puolestaan vähintään 80 pistettä. Opiskelija 1 saa tehtyä matematiikan tehtäviä tunnissa 20 pisteen arvosta ja opiskelija 2 5 pisteen arvosta samassa ajassa. Äidinkielessä opiskelija 1 saa 10 pistettä yhdessä tunnissa tehdyistä tehtävästä, kun taas opiskelija 2 saa tehtävästä 15 pistettä. Kerätään ongelmasta ehtolauseet ja objektifunktio.

$$20x_1 + 10x_2 \geq 60$$

$$5x_1 + 15x_2 \geq 80$$

$$x_1 + x_2 = z$$

kun $x_1 \geq 0, x_2 \geq 0$.

Muodostetaan yhtälöistä matriisi taulukko ja etsitään sen transpoosimatriisi.

$$\begin{bmatrix} 20 & 10 & 60 \\ 5 & 15 & 80 \\ 1 & 1 & 1 \end{bmatrix}^T = \begin{bmatrix} 20 & 5 & 1 \\ 10 & 15 & 1 \\ 60 & 80 & 1 \end{bmatrix}$$

Rakennetaan uudet epäyhtälöt saadun matriisitaulukon perusteella.

$$20y_1 + 5y_2 \leq 1$$

$$10y_1 + 15y_2 \leq 1$$
$$60y_1 + 80y_2 = p$$

kun $y_1 \geq 0, y_2 \geq 0$.

Lisätään yhtälöihin tarvittavat ylijäämämuuttujat. Tässä tapauksessa ylijäämämuuttujina toimivat x_1 ja x_2, joista myöhemmin saadaan myös alkuperäisen yhtälöiden muuttujien arvot.

$$20y_1 + 5y_2 + x_1 = 1$$
$$10y_1 + 15y_2 + x_2 = 1$$
$$60y_1 + 80y_2 = p$$

kun $y_1 \geq 0, y_2 \geq 0$

Muodostetaan ehdoista ja objektifunktiosta taulukko Simplex-algoritmin käyttöä varten.

	y_1	y_2	x_1	x_2	p	
R_1	20	5	1	0	0	1
R_2	10	15	0	1	0	1
R_3	-60	-80	0	0	1	0

Valitaan alimmalta riviltä negatiivisina arvo.

	y_1	y_2	x_1	x_2	p	
R_1	20	5	1	0	0	1
R_2	10	15	0	1	0	1
R_3	-60	-80	0	0	1	0

Valitaan nyt sarakkeelta luku, joka on eniten riippuvainen oman rivinsä kantaluvusta.

	y_1	y_2	x_1	x_2	p	
R_1	20	5	1	0	0	1
R_2	10	15	0	1	0	1
R_3	-60	-80	0	0	1	0

Kerrotaan rivi R_1 arvolla $\frac{1}{5}$ ja saadaan uusi rivi R_1', jolla käsittelyssä oleva luku muutetaan arvoon 1.

	y_1	y_2	x_1	x_2	p	
R_1	20	5	1	0	0	1
R_2'	$\frac{2}{3}$	1	0	$\frac{1}{15}$	0	$\frac{1}{15}$
R_3	-60	-80	0	0	1	0

Tämän jälkeen muokataan muita rivejä niin, että muuttuja y_2 löytyy vain riviltä R_2'. Eli nyt $R_1' = R_1 + (-5) \cdot R_2'$ ja $R_3' = R_3 + 80 \cdot R_2'$.

	y_1	y_2	x_1	x_2	p	
R_1'	$\frac{50}{3}$	0	1	-$\frac{1}{3}$	0	$\frac{2}{3}$
R_2'	$\frac{2}{3}$	1	0	$\frac{1}{15}$	0	$\frac{1}{15}$
R_3'	-$\frac{20}{3}$	0	0	$\frac{16}{3}$	1	$\frac{16}{3}$

Valitaan alimmalta riviltä negatiivisina arvo.

	y_1	y_2	x_1	x_2	p	
R_1'	$\frac{50}{3}$	0	1	-$\frac{1}{3}$	0	$\frac{2}{3}$
R_2'	$\frac{2}{3}$	1	0	$\frac{1}{15}$	0	$\frac{1}{15}$
R_3'	-$\frac{20}{3}$	0	0	$\frac{16}{3}$	1	$\frac{16}{3}$

Valitaan nyt sarakkeelta luku, joka on eniten riippuvainen oman rivinsä kantaluvusta.

	y_1	y_2	x_1	x_2	p	
R_1'	$\frac{50}{3}$	0	1	-$\frac{1}{3}$	0	$\frac{2}{3}$
R_2'	$\frac{2}{3}$	1	0	$\frac{1}{15}$	0	$\frac{1}{15}$
R_3'	-$\frac{20}{3}$	0	0	$\frac{16}{3}$	1	$\frac{16}{3}$

Kerrotaan rivi R_1' arvolla $\frac{3}{50}$ ja saadaan uusi rivi R_1'', jolla käsittelyssä oleva luku muutetaan arvoon 1.

	y_1	y_2	x_1	x_2	p	
R_1''	1	0	$\frac{3}{50}$	-$\frac{1}{50}$	0	$\frac{1}{25}$
R_2'	$\frac{2}{3}$	1	0	$\frac{1}{15}$	0	$\frac{1}{15}$
R_3'	-$\frac{20}{3}$	0	0	$\frac{16}{3}$	1	$\frac{16}{3}$

Tämän jälkeen muokataan muita rivejä niin, että muuttuja y_1 löytyy vain riviltä R_1''. Eli nyt $R_2'' = R_2' + (-\frac{2}{3}) \cdot R_1''$ ja $R_3'' = R_3' + \frac{20}{3} \cdot R_1''$.

	y_1	y_2	x_1	x_2	p	
R_1''	1	0	$\frac{3}{50}$	-$\frac{1}{50}$	0	$\frac{1}{25}$
R_2''	0	1	-$\frac{1}{25}$	$\frac{2}{25}$	0	$\frac{1}{25}$
R_3''	0	0	$\frac{2}{5}$	$\frac{26}{5}$	1	$\frac{28}{5}$

Ollaan tilanteessa, jossa alimmalla rivillä kaikki kantamuuttujien arvot ovat positiivisia.

	y_1	y_2	x_1	x_2	p	
R_1''	1	0	$\frac{3}{50}$	-$\frac{1}{50}$	0	$\frac{1}{25}$
R_2''	0	1	-$\frac{1}{25}$	$\frac{2}{25}$	0	$\frac{1}{25}$
R_3''	0	0	$\frac{2}{5}$	$\frac{26}{5}$	1	$\frac{28}{5}$

Taulukosta nähdään, että p saa arvon $\frac{28}{5}$ ja se myös vastaa muuttujan z arvoa.

	y_1	y_2	x_1	x_2	p	
R_1''	1	0	$\frac{3}{50}$	-$\frac{1}{50}$	0	$\frac{1}{25}$
R_2''	0	1	-$\frac{1}{25}$	$\frac{2}{25}$	0	$\frac{1}{25}$
R_3''	0	0	$\frac{2}{5}$	$\frac{26}{5}$	1	$\frac{28}{5}$

Taulukosta saadaan myös ylijäämämuuttujien kautta muuttujille x_1 ja x_2 arvot. Muuttuja x_1 saa arvokseen $\frac{2}{5}$ ja x_2 saa arvokseen $\frac{26}{5}$. Saadut arvot voidaan sijoittaa alkuperäisiin yhtälöihin ja nähdään, että ne täyttävät kaikki ongelman ehdot

$$20x_1 + 10x_2 \geq 60$$

$$5x_1 + 15x_2 \geq 80$$

$$x_1 + x_2 = z$$

kun $x_1 \geq 0, x_2 \geq 0$. Sijoitetaan yhtälöihin arvot $x_1 = \frac{2}{5}$ ja $x_2 = \frac{26}{5}$

$$20 \cdot \frac{2}{5} + 10 \cdot \frac{26}{5} = 60 \geq 60$$

$$5 \cdot \frac{2}{5} + 15 \cdot \frac{26}{5} = 80 \geq 80$$

$$\frac{2}{5} + \frac{26}{5} = \frac{28}{5} = z.$$

Voimme esimerkeistä huomata, kuinka työlääksi lineaarisen ohjelmoinnin ongelmien ratkaiseminen menee jo muutamalla muuttujalla ja ehtolauseella. Esimerkiksi kohdan 3.2 kuljetusongelma on jo järkevää ratkaista tietokoneen avustuksella.

Luku 6

Lineaarisen ohjelmoinnin opetuksen tutkiminen

Vuonna 1969 Tom E. Salazar teki lopputyönsä Montanan yliopistolle lineaarisen ohjelmoinnin opetuksesta toisen asteen oppilaitoksessa. Työn tarkoitus oli pohtia lineaarisen ohjelmoinnin opetusta toisen asteen oppilaitoksessa, ja miten lineaarisen ohjelmoinnin ideat parhaiten pystyttäisiin esittämään pedagogisesti. Työ perustelee lineaarisen ohjelmoinnin opetusta toisella asteella sillä, että työmarkkinoilla on tarve työntekijöille, jotka osaavat tehdä matemaattista optimointia. Tarve perustuu siihen, että työntekijä kykenee laskemaan, miten tuottoa voidaan maksimoida ja kuluja minimoida. [6, s. 1]

Salazarin mukaan lineaarisen ohjelmoinnin matemaattiset menetelmät kannattaa esitellä geometristen esitysten kautta. Esimerkkinä hän käyttää sitä, että useamman ehtolauseen sisältävän ongelman voi ratkaista visuaalisesti. Tutkielmassa myös perustellaan lineaarisiin ongelmiin jäämistä sillä, että rajoittamalla ehtolauseet lineaarisiksi ongelmien ratkaiseminen pysyy yksinkertaisena. Ongelmien pysyminen lineaarisina mahdollistaa niiden ratkaisemisen myös työkaluilla, jotka on suunniteltu nimenomaan lineaaristen ongelmien ratkaisemiseen. [6, s. 2-3,12-16]

6.1 Lineaarisen ohjelmoinnin ymmärtäminen toisella asteella

Vuonna 2018 julkaistiin Utomon tekemä tutkimus, jonka kohteena olivat korkean profiilin opiskelijat. Tutkimuksessa korkean profiilin opiskelijoilla tarkoitettiin matematiikkaa hyvin osaavia oppilaita. Tutkimuksessa seurattiin heidän matemaattisen ymmärryksen kehitystä lineaarisen ohjelmoinnin ongelmien ratkaisemisessa. Tutkimus keskittyi analysoimaan oppilaiden oppimista Pirie-Kieren laatiman ymmärryksen tasojen mukaan.[8, s. 1]

Pirie-Kieren malli on Susan Pirien ja Thomas Kieren kehittämä malli, joka kuvaa matemaattista ymmärrystä. Mallin kaksiulotteinen esitys muodostuu kahdeksasta eri ymmärryksen tasosta

- Taso 1 Alkeistieto: Ensimmäinen taso kuvaa matemaattisen ymmärryksen lähtötasoa. Tässä tasossa oppilaalla on oppimisen kannalta kaikki tarpeellinen taustatieto.
- Taso 2 Kuvan tekeminen: Toisessa tasossa oppilas hyödyntää aiempaa tietoa uudella tavalla ja alkaa muodostaa itselleen asiasta uutta kuvaa.
- Taso 3 Kuvan omaksuminen: Edellisen tason avulla oppilas luo itselleen uuden kuvan asioista, ja näistä muodostuneiden hahmotusmallien avulla oppilas voi käsitellä tietoa ilman, että mallia tarvitsee rakentaa joka kerta uudestaan.
- Taso 4 Kunnolla huomaaminen: Oppilas kykenee manipuloimaan asiasta saamaansa kuvaa tai hän pystyy yhdistelemään eri kuvia keskenään niin, että niistä löytyy kontekstiin kuuluvia ominaisuuksia. Tässä vaiheessa oppilas huomaa, että hänen mielessään muodostamansa kuvan mallit toimivat.
- Taso 5 Pätevä muoto: Seuraavalla tasolla ymmärrys siirtyy siihen pisteeseen, että oppilas pystyy soveltamaan matemaattista ymmärrystään yleisesti.
- Taso 6 Havainnointi: Kun oppilaalla on käsitys matemaattisen teorian yleisestä toimivuudesta, hän pystyy esittämään ja jäsentelemään teorioita. Nyt oppilas pystyy havainnoimaan matematiikkaa ja soveltamaan siihen sisäistä tietoaan.
- Taso 7 Jäsentäminen: Oppilas on tässä tasossa tietoinen, miten teorian kokonaisuudet toimivat keskenään ja pystyy tekemään loogisia perusteluja väitteilleen.
- Taso 8 Keksiminen: Oppilas on sisäistänyt asian niin hyvin, että pystyy muodostamaan uusia kysymyksiä, joista voi muodostua uusia käsitteitä perustuen jo opittuun tietoon. [7, s. 170-171]

Utomon tutkimuksessa seurattiin indonesialaisessa toisen asteen koulussa kahden oppilaan lineaarisen ohjelmoinnin ymmärrystä. Tutkimukseen valitut oppilaat olivat yhdennellätoista luokalla, jossa oppilaiden ikä on 16-17 vuotta. Tutkimuksessa kerättiin tietoja paperimuotoisten kokeiden avulla, ja niiden lisäksi oppilaille tehtiin haastatteluja tukemaan kokeista löytyneitä tuloksia. Tutkimuksen alussa mitattiin oppilaiden lähtötaso, jonka perusteella oppilailla oli hyvät edellytykset oppia lineaarista ohjelmointia. [8, s. 3]

Oppilaita seuratessa osoittautui, että oppimisessa oli haasteita. Tehtävän tietojen kirjallisessa ilmaisussa oli haasteita, kun oppilas ei ymmärtänyt ongelmaa kunnolla. Koska oppilas ei sisäistänyt matemaattista tietoa tarpeeksi hyvin, hänen piti palata takaisin tutkimaan ongelmaa lähtötietojensa perusteella. Matemaattisen ymmärryksen kannalta on oleellista osata lähtötiedot tarpeeksi hyvin, ennen kun voidaan aloittaa uuden asian omaksuminen. [8, s. 8-9]

Tutkimustulosten mukaan oppilaiden matemaattinen ymmärrys kehittyi vaiheittain. Oppilaat lähtivät alkeistietotasolta ja kykenivät piirtämään lineaariset ehdot graafisesti sekä kirjaamaan ylös ongelman oleelliset tiedot. Toinen ja kolmas ymmärryksen taso täyttyivät, kun oppilaat kykenivät kirjoittamaan lineaarisen ohjelmoinnin esimerkkejä. Oppilaat pääsivät neljännelle tasolle, kun he kykenivät määrittelemään ehtolauseet ja objektifunktion maksimiarvon. [8, s. 9]

Utomo esittää, että vaikka oppilailla olisi hyvä matemaattinen osaaminen, heidän tulisi palata alkeistiedon tasolle, kun he lähestyvät uutta asiaa. Tätä perustellaan sillä, että oppilaiden alkeistiedot eivät olleet tarpeeksi korkealla tasolla ja he eivät vielä olleet omaksuneet lähtötietoja tarpeeksi hyvin lineaarisen ohjelmoinnin ongelmien ratkaisemista varten. Tämän takia oppilaille tulisi antaa tehtäviä, jotka kasvattavat heidän matemaattisen ymmärryksensä lähtötietojen tasoa. [8, s. 9]

Utomo korostaa tutkimuksessaan, että matematiikan opetuksessa opettajan tulisi antaa ensin sellaista materiaalia oppilaille, että se tukee oppilaan lähtötason tietojen ymmärrystä. Vasta tämän jälkeen siirrytään uuteen asiaan. Opetusprosessissa tulisi myös käyttää Pirie-Kieren ymmärryksen tasomallia niin, että voidaan seurata oppilaiden matemaattisen ymmärryksen kehitystä. [8, s. 9]

6.2 Tutkimuksia lineaarisen ohjelmoinnin opetustavoista

Eri opetustapoja on tutkittu paljon. Sen seurauksena on myös tutkimuksia, joissa opetettavaksi matemaattiseksi aiheeksi on valittu lineaarinen ohjelmointi. Kaksi tällaista tutkimusta on tehty Indonesiassa toisen asteen oppilaitoksen oppilaiden opettamisesta. Ensimmäinen artikkeli, joka on nimeltään "Implementation of comic on linear program material to increase mathematical understanding for students of XI grade senior high school", tutki nimensä mukaisesti, miten sarjakuvien käyttö opetuksessa vaikutti lineaarisen ohjelmoinnin oppimiseen. Toinen tutkimus, jonka otsikkona on "The Efficiency of Using Education Videos on the Linear Program Material as Observed in Vocational High School Students' Mathematical Communication Ability", käsittelee puolestaan opetusvideoiden käyttöä lineaarisen ohjelmoinnin opetuksessa.

Indonesiassa vuonna 2018 tutkittiin toisen asteen opiskelijoita sen suhteen, kuinka opetusvideoiden käyttö vaikutti oppilaiden matemaattiseen kykyyn kommunikoida lineaarisen ohjelmoinnin konsepteja. Tutkimus toteutettiin ottamalla kaksi luokkaa, joista toiselle näytettiin opetusvideoita aiheesta ja toinen luokka toimi kontrolliluokkana. Tutkimuksessa kerättyjen tietojen perusteella oppilaat oppivat tehokkaammin lineaarista ohjelmointia opetusvideoiden avulla. Tämän tutkimuksen perusteella vaikuttaisi siltä, että opetusvideoiden käyttö lineaarisen ohjelmoinnin opetuksessa innostaa ja auttaa oppilaita seuraamaan paremmin opetusta kuin perinteinen opetus, jota kontrolliluokalle annettiin. [9, s. 11, 16]

Indonesiassa tutkittiin myös vuonna 2019, miten sarjakuvien käyttö edesauttoi lineaarisen ohjelmoinnin oppimista toisella asteella. Tutkimuksessa data kerättiin havainnoimalla oppimista, kokeilla ja oppilaille tehdyillä kyselyillä opetussarjakuvan käytön jälkeen. Tutkimuksen mukaan sarjakuvien käyttö opetusmenetelmänä paransi oppilaiden oppimista jonkin verran. Oppilaiden aktiviteettitaso luokassa kasvoi opetussarjakuvan käytön takia. Tutkimuksessa uskottiin tämän johtuvan siitä, että oppilaat ovat kiinnostuneita uusista erilaisista oppimistavoista. [10, s. 1, 6]

Luku 7

Lineaarisen ohjelmoinnin tuominen lukioon

Lukion vuoden 2015 opetussuunnitelmassa lineaarisessa ohjelmoinnissa oleellinen kahden muuttujan epäyhtälö esitellään jo pitkän matematiikan toisella kurssilla Polynomifunktiot ja -yhtälöt (MAA2). Yhtälöryhmien ratkaiseminen puolestaan esitellään pitkän matematiikan kurssilla 4 Vektorit (MAA4). Pitkän matematiikan kurssin 6 sisällöstä löytyvät raja-arvot, jatkuvuus ja derivaatta. [13, s. 131-133]

Huomataan, että jo pelkästään lukion pitkän matematiikan pakolliset kurssit antavat meille lähtötiedot, joita tarvitaan lineaarisen ohjelmoinnin ongelmien ratkaisemiseen. Pakollisten kurssien lisäksi on valtakunnallisia kursseja, jotka edesauttavat lineaarisen ohjelmoinnin ymmärtämistä.

Lukion valtakunnallisiin syventäviin kursseihin kuuluu MAA11, joka käsittelee lukuteoriaa ja todistamista. Sen sisältöön kuuluvat niin geometrinen todistaminen kuin induktiotodistus. Kurssi MAA12 Algoritmit matematiikassa esittelee matemaattisia algoritmeja ja niiden toimintaa. [13, s. 135]

Vaikka lukion lyhyellä matematiikalla ei ole yhtä kattavaa matemaattista kokonaisuutta kurssien suhteen, monet asiat, jotka käydään läpi pitkässä matematiikassa, sisältyvät myös lyhyen matematiikan kokonaisuuteen. Ottaen huomioon lineaarisen ohjelmoinnin käytännön hyödyt opiskelijalle, jolla on kiinnostusta talouden alaa kohtaan, lineaarisen ohjelmoinnin oppimisesta on käytännön hyötyä jatko-opinnoissa.

Vuoden 2015 opetussuunnitelmassa yksi oleellinen osa matematiikan opetusta on opettaa erilaisten tietokoneohjelmien hyödyntämistä matematiikan tutkimisessa. [13, s. 129] Lineaarisen ohjelmoinnin opiskelu tässä suhteessa voisi olla hyvin monipuolinen tilaisuus tutustua erilaisiin työkaluihin ja niiden rajoituksiin matematiikan ongelman ratkaisussa. Tietokoneohjelmat pystyvät ratkaisemaan monimutkaisia ongelmia, mutta vaativat käyttäjiltä ongelmien syöttämistä sellaisessa muodossa, että ohjelma pystyy niitä lukemaan.

7.1 Ehdotus lineaarisen ohjelmoinnin syventävän lukiokurssin sisällöstä

Lineaarisen optimoinnin opetus lukiossa antaa opiskelijoille yksinkertaisen ja tehokkaan työkalun käytännön maailman ongelmien optimointiin. Lineaarinen optimointi on selkeästi sidoksissa oikeisiin ongelmiin, joten se antaa opiskelijalle käsityksen siitä, miten monimutkaiselta näyttävää matematiikkaa voidaan soveltaa käytännössä. Lineaarisen ohjelmoinnin ongelmien mallien rakentaminen auttaa myös hahmottamaan, miten yritysten ongelmia voidaan muuttaa matemaattiseen muotoon.

Jos lineaarisesta ohjelmoinnista pitäisi tehdä lukion kurssikuvaus, se voisi olla esimerkiksi seuraavanlainen.

Tavoitteet
Kurssin tavoitteena on, että opiskelija

- perehtyy lineaaristen yhtälöryhmien perusominaisuuksiin
- osaa ratkaista yksinkertaisia lineaarisen ohjelmoinnin ongelmia visuaalisesti
- tutustuu matemaattisiin tietokoneohjelmiin, jotka auttavat monimutkaisten ohjelmien ratkomisessa
- osaa muuttaa sanallisia ongelmia matemaattiseen muotoon
- syventää ymmärrystään siitä, miten algoritmit toimivat matemaattisessa ratkaisuprosessissa
- ymmärtää yleisellä tasolla, miten Simplex-menetelmä ratkaisee lineaarisen ohjelmoinnin ongelman.

Keskeiset sisällöt

- lineaariset epäyhtälöt
- useamman ehtolauseen kuvaukset
- yksinkertainen matemaattinen ohjelmointi
- Simplex-menetelmä.

7.2 Lineaarisen ohjelmoinnin opettaminen lukiossa

Suomen lukiossa matematiikan opetuksen tehtävänä on esitellä opiskelijalle matemaattisen ajattelun malleja ja matematiikan rakenteita sekä perusideoita. Oleellinen osa opetusta on myös kehittää opiskelijan matemaattista kieltä ja ilmiöiden mallintamista. Lukion opetussuunnitelma ohjaakin opetusta valitsemaan opiskelijoita kiinnostavia aiheita ja ilmiöitä, joista muodostettavia ongelmia ratkaistaan erilaisilla työtavoilla. [13, s. 129]

Lukiossa matematiikan opetuksen yksi oleellinen osa on järjestää se siten, että opiskelija tekee itse havaintoja ja esittää omien havaintojensa perusteella kysymyksiä, oletuksia ja määritelmiä. Lukio-opetuksessa korostetaan sitä, että opiskelija hahmottaisi matemaattisia käsitteitä ja niiden yhteyttä laajempiin kokonaisuuksiin. Yksi osa tätä opetusprosessia on käyttää kuvia ja välineitä, joilla tuetaan matemaattisen tiedon siirtymistä esitysmuodosta toiseen. Matematiikan opetuksen tulisi luoda myös yhteyksiä matematiikan ja arkielämän välille. [13, s. 129]

Teoreettisen matematiikan lisäksi opiskelijoiden tulisi myös oppia käyttämään matematiikkaa tukevia tietokoneohjelmistoja. Matematiikan ala sisältää laajan skaalan erilaisia ohjelmistoja, jotka on suunniteltu auttamaan matematiikan käsittelyä. [13, s. 129]

Lineaarisessa ohjelmoinnissa on paljon asioita, jotka ovat yhteensopivia suomalaisen lukion matematiikan opetuksen periaatteiden kanssa. Lineaarisen ohjelmoinnin ongelmia voidaan tutkia monella eri tavalla, ja se havainnollistaa sitä, miten eri matemaattiset lähestymistavat ovat yhteydessä toisiinsa. Lineaarisen ohjelmoinnin ongelmissa merkittävä asia on asioiden havainnointi ja havaintojen muuttaminen matemaattiseksi kieleksi. Kun koko lineaarisen ohjelmoinnin ongelman ratkaisuprosessi on käyty läpi, siihen on sisältynyt jonkin ilmiön tarkastelua, sen matemaattista mallintamista, oikeiden matemaattisten työkalujen valintaa ongelman ratkaisuun ja valittujen työkalujen soveltamista ongelman ratkaisemiseksi.

Lineaarisen ohjelmoinnin tutkiessa isoja ongelmia on myös hyvä huomata, että isot kokonaisuudet voidaan jakaa pienemmiksi paloiksi. Opiskelija voi yhden ongelman aikana huomata, kuinka monipuolisia taitoja voidaan vaatia yhden käytännön ongelman ratkaisemiseksi. Lineaarinen ohjelmointi antaa myös hyvän käsityksen siitä, miksi tietokoneohjelmat eivät voi ratkaista kaikkia matemaattisia ongelmia, vaan ne vaativat ihmisen väliin tulkiksi. Tietokoneohjelman on lähes mahdotonta löytää kaikkia eri muuttujia tekstistä, johon on kirjattu ylös kaikki hyödyllinen ja turha tieto ongelman ratkaisemisen kannalta.

Yleisesti ottaen voisi olettaa, että lineaarinen ohjelmointi auttaa yhdistämään useita matemaattisia käsitteitä käytännön ilmiöihin. Lineaarisen ohjelmoinnin opetusta on myös helppo laajentaa tai supistaa sen mukaan, miten opiskelijat omaksuvat tietoa. On helppo esittää erilaisia tapoja ratkaista yksinkertaisia lineaarisen ohjelmoinnin ongelmia ja samalla pystyä osoittamaan, miksi suurin osa näistä ratkaisutavoista ei enää toimi laajemmissa ongelmissa. Kun siirrytään laajoihin lineaarisen ohjelmoinnin ongelmiin, on myös helppo näyttää, miksi mekaaninen laskeminen kannattaa siirtää tietokoneen tehtäväksi.

7.3 Ohjelmia lineaarisen ohjelmoinnin ongelmien ratkaisemiseen

Lineaarisen ohjelmoinnin ongelmia voidaan ratkaista monilla eri tietokoneohjelmilla. Vaihtoehtoina on ilmaisia ja maksullisia sovelluksia, niin avoimeen kuin suljettuun lähdekoodiin perustuvia ratkaisuja. Ohjelmien hyödyntäminen vaatii kuitenkin sitä, että lineaarisen ohjelmoinnin ongelma on muutettu numeeriseen muotoon ja ongelma voidaan syöttää ohjelmaan, joka etsii ongelmalle ratkaisun.

Yksi ohjelma, joka tarjoaa lineaaristen ohjelmoinnin ongelmien ratkaisijan, on MATLAB, jonka optimointityökalupakkiin kuuluu lineaaristen ohjelmoinnin ongelmien ratkaisija. MATHLAB:issa ongelma voidaan syöttää ratkaisijaan taulukkomuodossa, eli ongelma

$$x_1 + x_2 \leq 2$$
$$x_1 + 3x_2 \leq 5$$
$$2x_1 + 2x_2 \leq 6$$
$$3x_1 + 4x_2 \leq 7$$

jossa objektifunktio

$$100x_1 + 200x_2 = z$$

tulee antaa ohjelmalle niin, että kerätään taulukkoon A kaikki muuttujien arvot oikeille kohdille syötteeksi.

```
A =
[1 1
1 3
2 2
3 4]
```

kerätään kantaratkaisut taulukkoon b

```
b =[2 5 6 7]
```

annetaan vielä objektifunktion arvot taulukkoon f

```
f =[100 200]
```

ja ongelma ratkaistaan komennolla

```
[x,fval]=linprog(f,A,b)
```

jonka jälkeen MATLAB antaa tulosteena tiedon siitä, löytyykö optimaalinen vastaus ja mikä se on. [14]

Avoimeen lähdekoodiin perustuvia erilaisia Simplex-algoritmia hyödyntäviä ohjelmia löytyy ohjelmavarasto-Internetsivuilta kuten GitHubista. Näitä ohjelmia on tehty eri ohjelmointikielillä kuten Pythonilla tai Javalla. Internetistä löytyvien ohjelmien hyödyntäminen vaatii kuitenkin ohjelman koodin tarkastamista, koska avoin lähdekoodi ei takaa sitä, että ohjelma olisi toimiva ja turvallinen.

7.4 Lineaarisen ohjelmoinnin opettaminen projektipohjaisella opetuksella

Projektipohjainen oppiminen pohjautuu ideaan, että oppilaat oppivat käsittelemällä todellisia ja monimutkaisia ongelmia. Opetus yleensä lähtee liikkeelle siitä, että oppilas esittää kysymyksiä opettajan antaman alkulähtökohdan perusteella. Oppilas pyrkii sitten löytämään vastaukset aloittamalla tutkimusprojektin kyseiseen asiaan.[11, s. 100]

Samoin kuin ongelmalähtöinen oppiminen, projektipohjainen oppiminen pyrkii opettaman sellaisten monimutkaisten ongelmien kautta, joihin ei ole olemassa yhtä vastausta. Prosessi alkaa siitä, että selvitetään, mitä oppilas tietää ja mitä hän haluaa tietää. Projektilähtöisen oppimisen ideana on oppia käsittelemällä samaa aiheita uudestaan ja uudestaan lisäämällä siihen jatkuvasti uutta asiaa.[11, s. 100-101]

Vuonna 2016 julkaistussa artikkelissa "Exploring the Effects of Project-based Learning in Secondary Mathematics Education" Vicki-Lynn Holmesin ja Yooyeun Hwangin tutkimustulokset viittaavat siihen, että projektilähtöinen opetus oli tehokasta oppilaiden lähtötasosta riippumatta. Heidän mukaansa silloin, kun oppilaat uskovat ymmärtävänsä matematiikkaa, heistä tulee motivoituneempia sen oppimisessa. Projektilähtöisen oppimisen uskotaan toimivan hyvin myös sen takia, että se nojaa pienryhmissä työskentelyyn. [12, s. 460-461]

Koska lineaarisessa ohjelmoinnissa käsitellään laajoja reaalimaailman ongelmia, lukiossa lineaarisen ohjelmoinnin kurssia voisi opettaa projektipohjaisen opetuksen avulla. Lineaarisen ohjelmoinnin ongelmien tarkastelun voisi aloittaa tämän tutkielman ensimmäisen esimerkin tavoin ottamalla käsittelyyn jonkin hyvin yksinkertaisen ongelman ja kasvattamalla sitä pikkuhiljaa niin, että lopulta tarvitaan jo tietokoneohjelmia ongelman ratkaisuun. Alkuperäistä ongelmaa ei välttämättä tarvitse edes muuttaa, vaan siihen voi ajan myötä lisätä lisää ehtoja.

Käytännössä projektin eteneminen voisi tapahtua seuraavissa vaiheissa.

- Vaihe 1: Aiheen valinta ja pohtiminen, miten se soveltuu lineaarisen ohjelmoinnin ongelmiin
- Vaihe 2: Tutkitaan, minkälaisia ehtoja kyseisestä projektista voisi löytyä
- Vaihe 3: Mitä asioita projektissa voisi optimoida
- Vaihe 4: Mallin rakentamista yksinkertaisista ja monimutkaisemmista lineaarisen ohjelmoinnin ongelmista
- Vaihe 5: Ratkaistaan yksinkertaisia ongelmia visuaalisesti
- Vaihe 6: Ratkaistaan monimutkaisempia ongelmia Simplex-menetelmällä
- Vaihe 7: Tutkitaan sitä, miten voidaan tietokoneohjelmien avulla ratkaista ongelmia, jotka ovat liian monimutkaisia käsin laskettavaksi.

Kurssin aikana voidaan myös tarpeen vaatiessa käydä läpi teoriaa ja muita tehtäviä täydentämään asioita, jotka eivät tule esille projektin tekemisen kautta.

Kirjallisuutta

[1] George B. Dantzig *INDUCTIVE PROOF OP THE SIMPLEX METHOD*, 1st ed. The RAND Corporation, Sonto Monica, California, 1960

[2] George B. Dantzig *Linear Programming and Extensions*, 1st ed. Princeton University Press Princeton, New Jersey, 1963

[3] George B. Dantzig, Mukund N. Thapa *Linear Programming 1: Introduction* Springer, ProQuest Ebook Central, 1997

[4] Saul I. Gass *Linear Programming Methods and Appliocations*, 4th ed. McGraw-Hill Book Company, New York, 1975

[5] Ping-Qi PAN *Linear Programming Computation.* Springer-Verlag, Berlin, Heidelberg, 2014

[6] Tom Eufrancio Salazar *Linear programming for the high school student*, Graduate Student Theses Dissertations, & Professional Papers, 1969. Viitattu [04.09.2019]. Saatavissa:
https://scholarworks.umt.edu/etd/5534

[7] Susan Pirie, Thomas Kieren *Growth in Mathematical Understanding: How Can We Characterise It and How Can We Represent It?* Educational Studies in Mathematics 26, no. 2 (1994): s. 165-190.

[8] Utomo, TA Kusmayadi, I Pramudya *High Profile Students' Growth of Mathematical Understanding in Solving Linier Programing Problems.* Journal of Physics: Conference Series 1008, no. 1 (2018): 012070.

[9] Zakiah Laela, Asy Syifa Nurul Saomi, Rita Syara, Wahyu Hidayat, Heris Hendriana *The Efficiency of Using Education Videos on the Linear Program Material as Observed in Vocational High School Students' Mathematical Communication Ability* Journal Of Educational Experts (JEE) 1, no. 1 (2018): s. 11.

[10] S. Setiyani *Implementation of Comic On Linear Program Material to Increase Mathematical Understanding for Students of XI Grade Senior High School.* Journal of Physics: Conference Series 1157, no. 3 (2019): 032083.

[11] Nadia Behizadeh *Enacting Problem-Posing Education Through Project-Based Learning.* The English Journal 104, no. 2 (2014): s. 99-104.

[12] Vicki-Lynn Holmes, Yooyeun Hwang *Exploring the Effects of Project-based Learning in Secondary Mathematics Education.* The Journal of Educational Research 109, no. 5 (2016): s. 449-463.

[13] LOPS (2015). *Lukion opetussuunnitelman perusteet 2015*, Opetushallitus 27.10.2015. Viitattu [04.09.2019]. Saatavissa:
https://www.oph.fi/fi/tilastot-ja-julkaisut/julkaisut/lukion-opetussuunnitelman-perusteet-2015

[14] The MathWorks, Inc. *linprog, Solve linear programming problems*, Viitattu [05.09.2019]. Saatavissa:
https://se.mathworks.com/help/optim/ug/linprog.html